

```
1 # R code 2
2
3 # Pino Trogu
4 # School of Design -- San Francisco state University
5 # DATA? TA-DA!
6 # Data Viz Workshop 2019-02-20
7 # CA 260 - 4:00 - 6:00 PM
8 # ALL CODE BELOW PROVIDED "AS IS"
9
10 # RStudio
11 # in Environment pane, Import Dataset > from Text file
12 # find and select file named "breCan.csv"
13 # note that only 34 states are included in the list -- to simplify the exercise, I deleted any state
... for which data was missing in any of the categories (columns)
14 # make sure to check heading (YES) button when importing
15 # otherwise R will create a new "V1, V2, etc" header
16 # and your first true data row will end up having text in it (your real header info)
17 # then you will get an error because R expects numerical values in that row instead of alphabetical
... values
18
19 # run summary to get an overview of the data
20 # type the following and "run" code. place cursor on or select code line and click "Run" button in
... control bar menu above
21 # info will appear in console pane at bottom
22 summary(breCan)
23
24 # PLOT BARS BY CANCER INCIDENCE RATE
25 # *****
26 # *****
27 # start plot bars with a few state names (default)
28 # plot for cancer incidence -- yearly average per 100K female pop. Period: 2011-2015
29
30 # this uses the data in the column named "inc11_15"
31 # the names.arg argument adds the names of the states from the states column
32 # the main argument places a title above the chart
33 # run code by fully selecting the line of code or placing the cursor anywhere on the line and
... clicking "Run" button in menu bar
34
35 barplot(breCan$inc11_15,
36         names.arg=breCan$state,
37         main="Breast cancer incidence per 100K F pop. 2011-2015")
38
39 # notice that there is not enough room for all the names
40 # enlarging the plot pane creates more room but it's a fuzzy process and there is still not enough
... room for all the states
41
42 # ADD VERTICAL NAMES
43 # *****
44 # *****
45 # start plot bars will all the state names (vertical labels)
46
47 # to fit all the names, we'll make the labels vertical (srt = 90 degrees)
48 # by first adding the names argument with an empty value (nothing between the quotation marks)
49
```

```
50 barplot(breCan$inc11_15,
51         names.arg="",
52         main="Breast cancer incidence per 100K F pop. 2011-2015")
53
54 # next we create the "midpts" variable
55 # the reference point to where the labels will be positioned
56 # I believe it's the middle of the base of each bar
57 # this will not draw anything, but a midpoint series will be created and appear in the dataset pane
58
59 midpts <- barplot(breCan$inc11_15,
60                  names.arg="",
61                  main="Breast cancer incidence per 100K F pop. 2011-2015")
62
63 # this creates a series of coordinate points (locations, not visible) that will be used to position
... the labels
64 # text will add the labels in the x and y position relative to midpoint (play around with those
... values -- to start, I used +0.5 for X and -2 for Y)
65 # srt defines the angle (90 degrees in this case)
66
67 text(x=midpts+.5,
68      y=-2,
69      breCan$state,
70      cex=0.5,
71      srt=90,
72      xpd=TRUE,
73      pos=2)
74
75 # you can play with the x and y values but redraw the graph from barplot command first (line 45), and
... run midpoints and everything else again (45 through 66) otherwise the new labels will overlap the old
76
77 # ADD NAMES AT A 45 DEGREE ANGLE
78 # *****
79 # *****
80 # turn labels at 45 degrees to make them more legible and elegant
81
82 # plot bars first again
83 barplot(breCan$inc11_15,
84         names.arg="",
85         main="Breast cancer incidence per 100K F pop. 2011-2015")
86
87 # set midpoints again
88 midpts <- barplot(breCan$inc11_15,
89                  names.arg="",
90                  main="Breast cancer incidence per 100K F pop. 2011-2015")
91
92 # add text -- change x and y values as needed, cex is size of type, srt = 45 degrees
93 # xpd has to do with the working area for the graph (just leave as is)
94 # pos 2 means text aligned right
95 text(x=midpts+1.5,
96      y=-4,
97      breCan$state,
98      cex=0.5,
99      srt=45,
100     xpd=TRUE,
```

```
101     pos=2)
102
103
104 # SORT BARS BY CANCER INCIDENCE, NOT ALPHA BY STATE NAME
105 # *****
106 # *****
107 # plot bar chart for incidence again (but sort by incidence, not alphabetically by state)
108
109 # plot bars first again
110 barplot(breCan$inc11_15,
111         names.arg="",
112         main="Breast cancer incidence per 100K F pop. 2011-2015")
113
114 # create new virtual dataset where "inc11_15" is sorted
115 # a new dataset called "sortByInc" -- the name can be anything -- will appear in dataset pane
116 sortByInc <- breCan[order(breCan$inc11_15), ]
117
118 # set midpoints again and plot bars
119 # names will be state in sortByInc
120 # note that now you have to use sortByInc before dollar sign (instead of breCan)
121 # because the sorting will change, the bars will be redrawn
122 midpts <- barplot(sortByInc$inc11_15,
123                   names.arg="",
124                   main="Breast cancer incidence per 100K F pop. 2011-2015")
125
126 # add text (labels) -- play with x and y values to position labels as you like them
127 # make sure to again use sortByInc before $state, otherwise the state names will still be
... alphabetical and everything will be scrambled
128 text(x=midpts+1,
129      y=-4,
130      sortByInc$state,
131      cex=0.5,
132      srt=45,
133      xpd=TRUE,
134      pos=2)
135
136
137 # INVERT SORTING FROM HIGHEST TO LOWEST (LEFT TO RIGHT)
138 # *****
139 # *****
140 # plot again but invert sorting (cancer incidence from high to low)
141
142 # plot bars again
143 barplot(breCan$inc11_15,
144         names.arg="",
145         main="Breast cancer incidence per 100K F pop. 2011-2015")
146
147 # new sorted table with decreasing = TRUE
148 # note you must keep the comma after "TRUE)"
149 sortByInc <- breCan[order(breCan$inc11_15,
150                           decreasing = TRUE), ]
151
152 # new midpoints to pick up the new label positions
153 midpts <- barplot(sortByInc$inc11_15,
```

```
154         names.arg="",
155         main="Breast cancer incidence per 100K F pop. 2011-2015")
156
157 # new labels
158 text(x=midpts+.5,
159      y=-4,
160      sortByInc$state,
161      cex=0.5,
162      srt=45,
163      xpd=TRUE,
164      pos=2)
165
166 # in the plot pane, export plot to PDF
167
168
169 # PLOT BARS BY MORTALITY RATE
170 # *****
171 # *****
172 # plot bar chart for cancer mortality -- yearly average per 100K female pop. Period: 2012-2016
173 # repeat and experiment with midpoints and text as in the examples for incidence
174 # note that now we add the title of the graph (main) in the midpoint code, not the barplot code
175
176 barplot(breCan$mort12_16,
177         names.arg="")
178
179 midpts <- barplot(breCan$mort12_16,
180                  names.arg="",
181                  main="Breast cancer mortality per 100K F pop. 2012-2016")
182
183 text(x=midpts+.5,
184      y=-2,
185      breCan$state,
186      cex=0.5,
187      srt=45,
188      xpd=TRUE,
189      pos=2)
190
191
192 # SORT BARS BY MORTALITY, NOT ALPHA BY STATE
193 # *****
194 # *****
195 # plot bar chart for mortality again (but sort by rate, not alphabetically by state)
196 # also adjust label position -- x and y distance from base of bars
197 # use the name sortByMort for the sorting
198
199 barplot(breCan$mort12_16,
200         names.arg="")
201
202 sortByMort <- breCan[order(breCan$mort12_16), ]
203
204 midpts <- barplot(sortByMort$mort12_16,
205                  names.arg="",
206                  main="Breast cancer mortality per 100K F pop. 2012-2016")
207
```

```
208 text(x=midpts+1.3,
209       y=-.5,
210       sortByMort$state,
211       cex=0.5,
212       srt=45,
213       xpd=TRUE,
214       pos=2)
215
216
217 # INVERT SORTING -- NOW FROM HIGH MORT. TO LOW MORT.
218 # *****
219 # *****
220 # plot again but reverse sorting (high rate to low)
221 # add decreasing = TRUE to sort from high to low
222 # note you must keep the comma after "TRUE)"
223
224 barplot(breCan$mort12_16,
225         names.arg="")
226
227 sortByMort <- breCan[order(breCan$mort12_16,
228                             decreasing = TRUE), ]
229
230 midpts <- barplot(sortByMort$mort12_16, names.arg="",
231                  main="Breast cancer mortality per 100K F pop. 2012-2016")
232
233 text(x=midpts+1.3, y=-.5, sortByMort$state, cex=0.5, srt=45, xpd=TRUE, pos=2)
234
235 # export plot to PDF
236
237
238 # *****
239 # *****
240 # SCATTERPLOT INCIDENCE (X-AXIS) AND MORTALITY (Y-AXIS)
241 # *****
242 # *****
243
244 # plot scatterplot of incidence rates for the period 2011-2015
245 # and mortality rates for the period 2012-2016
246 # per 100K population (yearly average by state)
247
248 # plot using simply the original file name
249 # this gives a complete matrix of all plots using any two variables found in your dataset (13 columns
... -- 13 x 13 matrix)
250 plot(breCan)
251
252 # it's small to see details but if you export the plot to PDF and open it in Illustrator or Preview
... you will be able to read all column names (along diagonal) -- move across and vertically from a pair
... of names to locate a specific plot
253
254 # or plot only a range of columns (we need cols. 1 through 3 )
255 plot(breCan[1:3])
256
257 # run summary of data set
258 summary(breCan)
```

```
259
260
261 # SCATTERPLOT WITH TWO VARIABLES
262 # *****
263 # *****
264 # plot just two variables: cancer incidence and cancer mortality
265 plot(breCan$inc11_15,
266       breCan$mort12_16)
267
268 # *****
269 # *****
270 # plot same but add a title (top) and subtitle (bottom) using the title line
271 # note the nested parentheses
272 plot(breCan$inc11_15,
273       breCan$mort12_16,
274       title(main="Breast cancer incidence 2011-2015 and mortality 2012-2016 by state",
275             sub="Yearly average per 100K F pop. -- some states not shown")
276       )
277
278 # add names of states to each dot, using the state column in the third argument -- the first and the
... second arguments define the location (X and Y by default, same as the dots)
279 # labels are centered on dots
280 # play with cex to change size of label text
281 text(breCan$inc11_15,
282       breCan$mort12_16,
283       breCan$state,
284       cex=0.5)
285
286 # add some grid lines to the plot ("dotted" or "solid")
287 # for details, see https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/grid.html
288 grid(nx = NULL,
289       ny = NULL,
290       col = "lightgray",
291       lty = "solid",
292       lwd = par("lwd"),
293       equilog = TRUE)
294
295
296 # USE DATA FROM symHack COLUMN (ALL DATA = 10)
297 # TO DRAW CIRCLES (FANCY DOTS) ALL THE SAME SIZE
298 # BUT CONTROL HOW BIG THEY ARE USING THE INCHES SETTING
299 # *****
300 # *****
301 # plot circles again but specify size from symHack
302 # all cells are same (10) so all circles will be same
303 symbols(breCan$inc11_15,
304         breCan$mort12_16,
305         circles=breCan$symHack)
306
307 # way too big!
308 # specify size (inches)
309 symbols(breCan$inc11_15,
310         breCan$mort12_16,
311         circles=breCan$symHack,
```

```
312     inches=0.2)
313
314 # even smaller
315 symbols(breCan$inc11_15,
316         breCan$mort12_16,
317         circles=breCan$symHack,
318         inches=0.05)
319
320 # add labels to each circle (name of each state)
321 text(breCan$inc11_15,
322      breCan$mort12_16,
323      breCan$state,
324      cex=0.5)
325
326 # export plot to PDF
327
328 # for fun, run regression line to see trend in the data
329 symbols(breCan$inc11_15,
330         breCan$mort12_16,
331         circles=breCan$symHack,
332         inches=0.05)
333
334 scatter.smooth(breCan$inc11_15, breCan$mort12_16)
335
336 # add labels to each circle (name of each state)
337 text(breCan$inc11_15,
338      breCan$mort12_16,
339      breCan$state,
340      cex=0.5)
341
342 # *****
343 # *****
344 # USE DATA FROM POPULATION COLUMN IN DATASET
345 # TO PROPORTION CIRCLES BASED ON STATE POPULATION SIZE
346 # THIS ADDS A THIRD VARIABLE ON THE SAME PLOT (POP. -- AN IMAGINARY "Z" AXIS)
347 # will use to show the relative size of states' total population
348 # use with caution as larger circles tend to skew perception towards them
349 # however the point of this graph is "rate per 100,000 female population", not total population
350 # but test what happens anyway
351 # *****
352 # *****
353
354 # plot using specific shapes -- circles in this case, but they could be squares
355 # the size of the circles (bubbles) is based on state population
356 # note: proportions will not be correct, but will fix that later
357 # here we will leave title and subtitle out for simplicity
358 symbols(breCan$inc11_15,
359         breCan$mort12_16,
360         circles=breCan$pop)
361
362 # add names of states
363 text(breCan$inc11_15,
364      breCan$mort12_16,
365      breCan$state,
```

```
366     cex=0.5)
367
368
369 # MAKE CIRCLES PROPORTIONAL BY SQUARING (SQUARE ROOT) THE POPULATION QUANTITIES
370 # *****
371 # *****
372
373 # notice how in the graph California (about 40M pop.) looks way much bigger than New York (about 20M
... pop.)
374 # it should look only about twice as big
375 # so the the bubbles are not proportional
376 # because they are built by using the data for radius (or diameter) instead as for area
377 # use sqrt (sqrt -- square root) to correctly size each bubble, proportional to each other
378 # also set inches dimension for the maximum circle size
379 symbols(breCan$inc11_15,
380         breCan$mort12_16,
381         circles=sqrt(breCan$pop),
382         inches=0.25)
383
384 # add names of states
385 text(breCan$inc11_15,
386      breCan$mort12_16,
387      breCan$state,
388      cex=0.5)
389
390 # add some grid lines ("dotted" or "solid") to the plot
391 # for details, see https://stat.ethz.ch/R-manual/R-devel/library/graphics/html/grid.html
392 grid(nx = NULL,
393      ny = NULL,
394      col = "lightgray",
395      lty = "solid",
396      lwd = par("lwd"),
397      equilog = TRUE)
398
399 # export plot to PDF and clean up and modify design in Illustrator
400
401 # END OF R FILE
```